INTERSPEECH 2017
August 20–24, 2017, Stockholm, Sweden

Generative Adversarial Network-based Postfilter for STFT Spectrograms

Takuhiro Kaneko[1], *Shinji Takaki*[2], *Hirokazu Kameoka*[1], *Junichi Yamagishi*[2]

[1]NTT Communication Science Laboratories, NTT Corporation, Japan
[2]National Institute of Informatics, Japan

{kaneko.takuhiro,kameoka.hirokazu}@lab.ntt.co.jp, {takaki,jyamagis}@nii.ac.jp

Abstract

We propose a learning-based postfilter to reconstruct the high-fidelity spectral texture in short-term Fourier transform (STFT) spectrograms. In speech-processing systems, such as speech synthesis, conversion, enhancement, separation, and coding, STFT spectrograms have been widely used as key acoustic representations. In these tasks, we normally need to precisely generate or predict the representations from inputs; however, generated spectra typically lack the fine structures that are close to those of the true data. To overcome these limitations and reconstruct spectra having finer structures, we propose a generative adversarial network (GAN)-based postfilter that is implicitly optimized to match the true feature distribution in adversarial learning. The challenge with this postfilter is that a GAN cannot be easily trained for very high-dimensional data such as STFT spectra. We take a simple divide-and-concatenate strategy. Namely, we first divide the spectrograms into multiple frequency bands with overlap, reconstruct the individual bands using the GAN-based postfilter trained for each band, and finally connect the bands with overlap. We tested our proposed postfilter on a deep neural network-based text-to-speech task and confirmed that it was able to reduce the gap between synthesized and target spectra, even in the high-dimensional STFT domain.

Index Terms: postfilter, deep neural network, generative adversarial network, statistical parametric speech synthesis

1. Introduction

The aim with many speech-processing systems, including speech synthesis, conversion, enhancement, separation, and coding, is to produce speech with quality indistinguishable from clean and real speech. However, the quality of synthesized or processed speech is usually not as good as that of real speech. In this paper, we address the problem of restoring spectro-temporal fine details of a synthetic speech signal to make it sound like real speech.

Many methods for statistical parametric speech synthesis and voice conversion tend to produce over-smoothed spectra, which often result in muffled and buzzy synthesized speech. This is caused by a side effect of assuming a particular form of loss function (e.g., mean squared error) or distribution (e.g., Gaussian) for parameter training of the acoustic model. Conventionally, postfiltering methods based on the global variance [1, 2] or modulation spectrum [3] have proved to be effective in improving the intelligibility of synthesized speech.

Speech enhancement and separation are typically carried out using a Wiener filter or time-frequency mask. While a time-frequency mask allows aggressive suppression of noise components, it can also over-suppress and damage speech components. A Wiener filter provides a conservative way of separating out a speech signal from a mixture signal so that the sum of the separated signals is ensured to be equal to the mixture; however, it often produces artifacts perceived as time-varying tones known as musical noise. To reduce artifacts or musical noise in processed speech, postprocessing methods using cepstral smoothing techniques have been proposed [4].

The limitation of these postprocessing methods is that they rely on heuristics due to the difficulty of modeling the exact probability density of the spectrograms of real speech. This typically causes generated spectra to lack the fine structures that are close to those of the true data. Recently, learning-based postfilters have been proposed [5, 6]. These postfilters are optimized using a particular form of loss function or distribution. However, it is difficult to completely overcome the over-smoothing problem as long as a manually designed metric is used.

To overcome these limitations, we previously proposed [7] the use of a generative adversarial network (GAN) [8], which makes it possible to generate random samples following the underlying data distribution without the need for the explicit form of its density, to construct a postfilter for acoustic-feature sequences generated using a deep neural network (DNN)-based text-to-speech (TTS) synthesizer. In that work, we discussed the effectiveness of our postfilter when applied to a sequence of low-dimensional vocoder parameters, such as the mel-cepstral features; however, its effectiveness when applied to a sequence of high-dimensional features, such as the short-time Fourier transform (STFT) spectra, has not been clarified.

Motivated by this background, in this paper, we propose a GAN postfilter that allows the handling of high-dimensional features, such as the STFT spectra, so that it can be applied to any speech-processing system (not limited to speech synthesis) that produces the spectrograms of speech. This is particularly useful and convenient because once a magnitude spectrogram is obtained, we can use phase-reconstruction algorithms [9] to reconstruct a time-domain waveform signal.

We also previously proposed a DNN-based TTS system that directly produces a sequence of STFT spectra in the hope of going beyond the limitation of the sound quality of vocoders [10]. In this paper, we investigate the application of our proposed postfilter to the STFT spectrograms generated using our DNN-TTS system. The experimental results revealed that the use of the proposed postfilter had a certain effect in reducing the gap between synthesized and target spectra, even in the high-dimensional STFT domain.

This paper is organized as follows. In Section 2, we explain the proposed GAN-based postfilter for STFT spectrograms. In Section 3, we explain how we used it in our DNN-based TTS system. We present the experimental results in Section 4 and summarize our findings in Section 5.

http://dx.doi.org/10.21437/Interspeech.2017-962

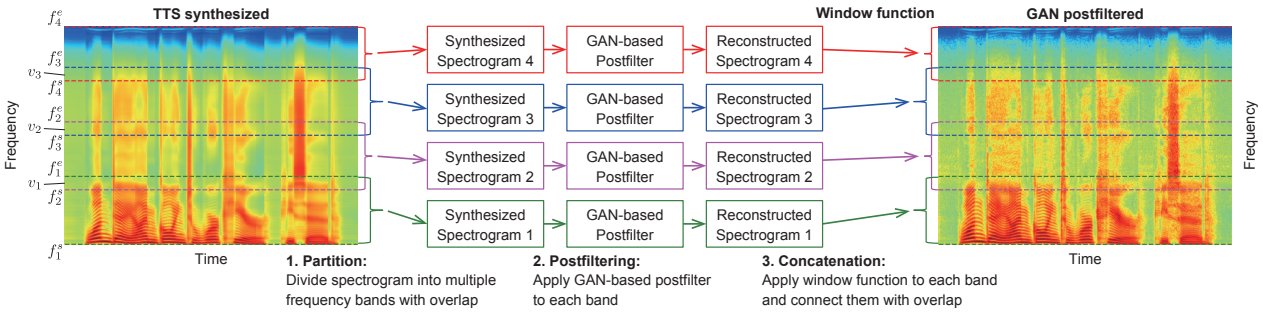

Figure 1: *System overview of proposed GAN-based postfilter for high-dimensional STFT spectrograms*

2. GAN-based postfilter

Our proposed postfilter is built upon the GAN-based postfilter that we previously proposed [7]. In this section, we first briefly review a GAN then explain the formulation for using a GAN as a postfilter for vocoder parameters. Next, we introduce our proposed postfilter that can be applied to high-dimensional STFT spectrograms.

2.1. GAN

A GAN [8] is a framework for estimating a generative model by an adversarial process, and the goal is to learn a generator distribution $P_G(x)$ that matches the true data distribution $P_{\mathrm{Data}}(x)$. It is composed of two networks. One is a generator G that maps noise variables $z \sim P_{\mathrm{Noise}}(z)$ to the data space $x = G(z)$. The other is a discriminator D that assigns probability $p = D(x)$ when x is sampled from the $P_{\mathrm{Data}}(x)$ and assigns probability $1 - p$ when x is sampled from the $P_G(x)$. The D and G play a two-player minmax game, and the GAN objective is written as

$$\min_G \max_D \mathbb{E}_{x \sim P_{\mathrm{Data}}(x)}[\log D(x)] + \mathbb{E}_{z \sim P_{\mathrm{Noise}}(z)}[\log(1 - D(G(z)))]. \quad (1)$$

This enables the D to find the binary classifier that gives the best possible discrimination between true and generated data and simultaneously enables the G to fit the $P_{\mathrm{Data}}(x)$. Both G and D can be trained using back-propagation.

2.2. GAN-based postfilter for vocoder parameters

By focusing on the functionality of a GAN that can implicitly optimize the G to match the true data distribution through adversarial learning, we previously proposed a GAN-based postfilter to reconstruct the true spectral texture generated from a vocoder [7]. We made three changes to the regular GAN architectures by using conditional, residual, and convolutional networks for postfiltering.

Conditional: Our goal is to reconstruct natural spectral texture x from synthesized spectral texture y and random noise z. To achieve this, we use a conditional GAN (CGAN) [11, 12], which is an extension of a GAN, where the G and D receive additional y data as input:

$$\min_G \max_D \mathbb{E}_{x,y \sim P_{\mathrm{Data}}(x,y)}[\log D(x, y)] + \mathbb{E}_{z \sim P_{\mathrm{Noise}}(z), y \sim P_y(y)}[\log(1 - D(G(z, y), y))]. \quad (2)$$

Here, z represents the stochastic fluctuation in reconstructing a natural spectral texture from a synthesized one.

Residual: To shorten the entire process of generating the spectral texture, we design the G as follows

$$G(x, y) = y + R(x, y), \quad (3)$$

where R represents residual texture [13].

Convolutional: Based on the observation that a spectral texture is structured in both time and frequency directions, we use convolutional architecture to determine the structure with a reasonably small number of parameters. In particular, we design the G as a fully convolutional network (FCN) [14] that allows input segments to take an arbitrary length.

2.3. GAN-based postfilter for STFT spectrograms

The STFT spectrogram is not only high dimensional but also has a different structure depending on the frequency bands, e.g., a clear harmonic structure can be observed in the low-frequency band, while randomness increases in the high-frequency band. This would make it difficult to estimate the spectrogram distribution with our naive GAN-based postfilter described in Section 2.2. To mitigate this problem, we propose converting a spectrogram on a band-by-band basis. Namely, we divide the spectrogram into multiple bands, reconstruct the individual bands using the GAN-based postfilter trained for each band, and finally concatenate them. The system overview is summarized in Figure 1. We describe the details of each step as follows.

Partition: We first divide the spectrogram into N frequency bands, each of which ranges from the f_i^s-th to f_i^e-th frequency, where N is the number of bands and $i = \{1, \ldots, N\}$. The overlap between the i-th and $i+1$-th bands is set at v_i, i.e., $v_i = f_i^e - f_{i+1}^s$. We use the overlap representation to smoothly concatenate the individual bands afterwards.

Postfiltering: We reconstruct the individual bands using the GAN-based postfilter trained for each band. The spectrogram in each band is not only lower dimensional but also has a more homogeneous structure than the entire spectrogram; therefore, we expect that it is easier to model.

Concatenation: To smoothly connect the reconstructed spectrograms, we apply a window function (e.g., hanning, hamming, or Blackman) to both ends of each band before connection, where the window width is $2v_i$ and half of the window function is applied to each end. This method works well, as shown in the reconstructed spectrogram in Figure 1. In preliminary experiments, we also tested a model in which the spectrograms are divided and connected without overlap. In this model, the reconstructed spectrogram tends to have discontinuity between the bands, causing a popping sound.

3. Application to speech synthesis

In this section, we describe how we used the proposed GAN-based postfilter in a DNN-based TTS system. The improved components include direct STFT-spectra prediction from text, postfiltering of the predicted STFT spectra, and waveform generation using enhanced STFT spectra and phase recovery.

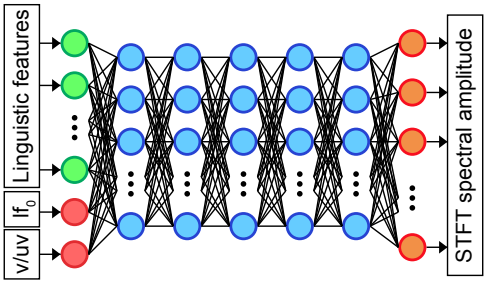

Figure 2: *DNN-based acoustic model for STFT spectra. In this model,* $\log F_0$ *(lf*$_0$*) and voiced/unvoiced (v/uv) values are used as input features as well as linguistic features.*

Direct STFT-spectra prediction from text: Figure 2 shows the acoustic model used in this study. This acoustic model directly predicts STFT spectra based on a feed-forward NN [10]. Also, in contrast to a conventional DNN-based acoustic model [15], we use F_0 information as input features as well as linguistic features to predict STFT spectra, which include harmonics derived from F_0. To use the benefit of directly using the STFT spectra, the Kullback-Leibler divergence (KLD)-based criterion [16, 17] is used for training a DNN effectively. According to our previous experiment [10], this DNN-based TTS system leads to better quality of synthetic speech than that generated from a system using a vocoder.

Postfiltering of predicted STFT spectrograms: The predicted STFT spectra from the acoustic model is enhanced by a signal-processing-based postfilter first [18], followed by the proposed GAN-based postfilter described in the previous section. We apply the signal-processing-based postfilter for enhancing spectral peaks of predicted spectral amplitudes as follows. 1) The predicted STFT spectral amplitudes are converted into linear-scale cepstrum vectors that have the same dimensions as the STFT amplitudes, 2) the postfilter is applied to the cepstrum vectors for peak enhancement, and 3) the cepstrum vectors after postfiltering are converted back into the spectral amplitudes. These modified STFT spectra are then used as input features for the proposed GAN-based postfilter.

Waveform generation based on phase recovery: The speech-waveform generated from postfiltered STFT spectral amplitudes is based on the well-known phase-recovery algorithm proposed by Griffin and Lim [9]. The algorithm consists of the following iterative steps.

1. Generate initial speech waveforms using the inverse STFT of predicted STFT spectral amplitudes A with or without a postfilter and random phase θ at each frame, followed by overlap-add operations.
2. Window the speech waveforms and apply STFT at each frame to obtain new spectral amplitudes $\widehat{A}$ and phase values $\widehat{\theta}$.
3. Replace the STFT $\widehat{A}$ with the original A at each frame.
4. Generate new speech waveforms using the inverse STFT of the original STFT spectral A and updated $\widehat{\theta}$ followed by overlap-add operations.
5. Go back to step 2 until convergence.

This framework may be considered as another type of DNN-based speech-synthesis system without a vocoder to avoid quality deterioration such as buzziness caused by using the vocoder. In this framework, the quality of synthetic speech totally relies on the prediction accuracy of the STFT spectra.

Table 1: *Network architectures for GAN-based postfilter*

Generator (**Input:** $F \times T$ spectrogram + $F \times T$ noise)
5×5 128 conv., ReLU + input spectrogram
5×5 256 conv., ReLU + input spectrogram
5×5 128 conv., ReLU + input spectrogram
5×5 1 conv.
Discriminator (**Input:** $F \times T_c$ spectrogram)
5×5 64 conv. $\downarrow$, LReLU
5×5 128 conv. $\downarrow$, BNorm, LReLU
5×5 256 conv. $\downarrow$, BNorm, LReLU
5×5 512 conv. $\downarrow$, BNorm, LReLU
1 fully connected, sigmoid

We observed that refinement of the amplitudes using a signal-processing-based postfilter improves the synthesized speech quality [10], but the filter is designed to enhance the peak frame-by-frame; hence, the characteristics of the STFT spectra, i.e., time-frequency dependency, are not appropriately considered. Therefore, the proposed GAN-based postfilter for the STFT spectra is expected to further improve the synthesized speech quality.

4. Experiments

4.1. Experimental conditions

The database that was provided for the Blizzard Challenge 2011 [19], which contains approximately 17 hours of speech data composed of 12,000 utterances, was used for the experiment. All data were down-sampled from 48 to 32 kHz. Two hundred sentences not included in the database were used as a test set. The speech data were windowed at a frame rate of 160 points (5 ms) to extract their frequency spectra with 1025 STFT points. The linguistic features for the DNN-based acoustic model were composed of 396 dimensions. The $\log F_0$ and voiced/unvoiced values were also used as input features of the DNN-based acoustic model. Five-hidden-layer feed-forward neural networks with a sigmoid based activation function were used for the acoustic model.

In the synthesis phase, we used the $\log F_0$ and voiced/unvoiced values predicted from a conventional vocoder-based system [20] as input features. In this conventional system, 49-dimensional bark-cepstral coefficients (plus the 0th coefficient) obtained from WORLD spectra, $\log F_0$, 25-dimensional band aperiodicity measures, their dynamic and acceleration coefficients, and voice/unvoiced values were modeled based on a DNN. Phoneme durations were estimated by a HMM-based speech synthesis system [21].

In the postfiltering phase, we divided the spectrogram into four frequency bands. We set the parameters as $(f_1^s, f_1^e) = (1, 320)$, $(f_2^s, f_2^e) = (257, 576)$, $(f_3^s, f_3^e) = (513, 832)$, and $(f_4^s, f_4^e) = (769, 1024)$. We connected the bands with the hamming-window function where the window width was $2v_i = 128$ $(i = 1, 2, 3)$. Table 1 lists the network architectures for the GAN-based postfilter. The symbol $\downarrow$ indicates down-sampling with stride 2. The terms *ReLU*, *LReLU*, and *BNorm* denote rectified linear unit [22], leaky rectified linear unit [23, 24], and batch normalization [25], respectively. As the input of the G, we used the $F \times T$ spectrograms and the same-sized noise where F is the frequency length and T is the frame length. The F was 320 for the first, second, and third bands and 256 for the fourth band. We designed the G as an FCN, so we could take inputs

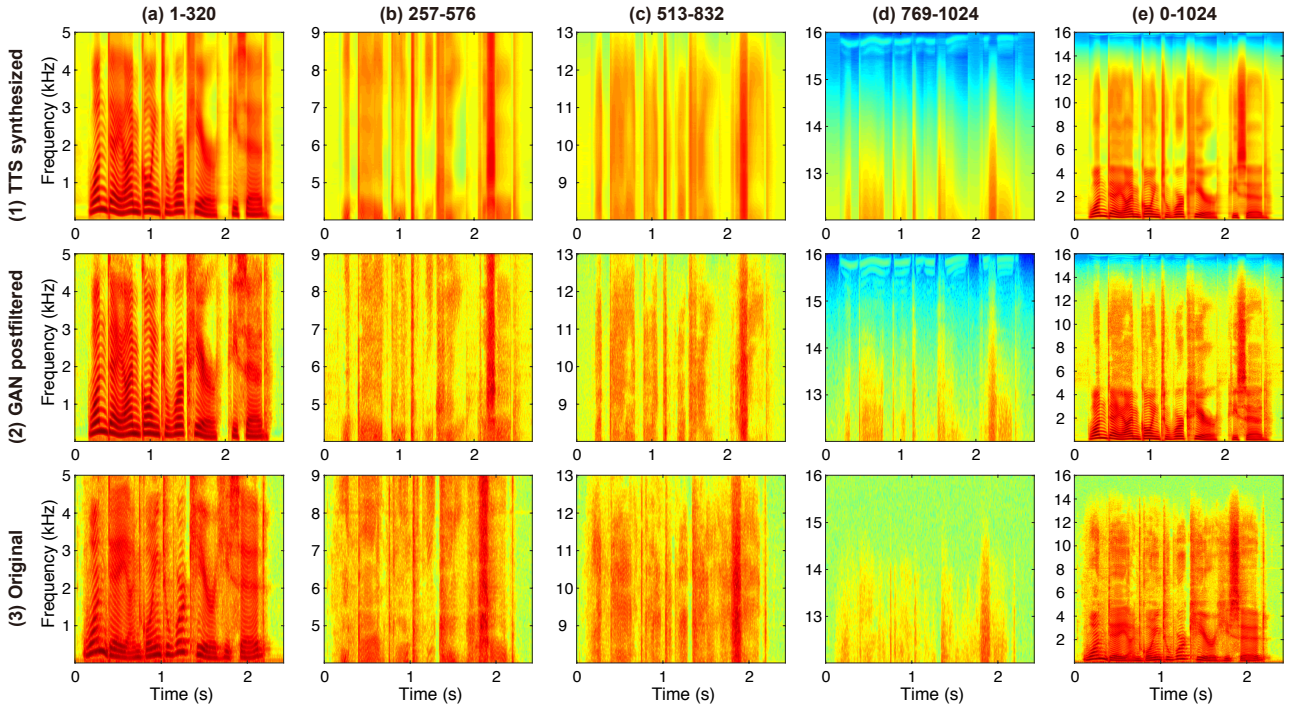

Figure 3: *Comparison of (1) TTS synthesized, (2) GAN postfiltered, and (3) original STFT spectrograms* [1]

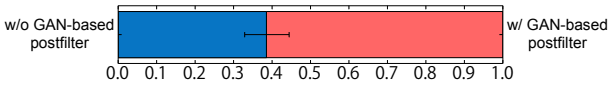

Figure 4: *Results from subjective listening test*

of arbitrary T. In the D, we used a fully connected architecture at the last layer, so we fixed the input size as $F \times T_c$ with $T_c = 64$. During pre-processing, we normalized spectrograms to zero-mean and unit-variance for each dimension using the training sets. We trained our postfilter using the Adam optimizer [26] with a minibatch of size 16. The learning rate was set to 0.0002 for the D and 0.001 for the G, and the momentum term was set to 0.5.

4.2. Objective evaluation

Figure 3 shows the comparison of the (1) TTS synthesized, (2) GAN postfiltered, and original STFT spectrograms. From (a) to (d), we show the spectrograms in the individual frequency bands. These results indicate that the proposed GAN-based postfilter can not only emphasize the harmonic structure but also reproduce the detailed structures that are similar to those in the original spectrograms from the over-smoothed synthesized spectrograms. Figure 3(e) shows all spectrograms. Our postfilter enables the individual frequency bands to be connected smoothly.

4.3. Subjective evaluation

We conducted a listening test to compare a DNN-based TTS system with the proposed GAN-based postfilter with another DNN-based TTS system without the postfilter. The listening test that we used is a normal preferences test. The test was conducted in acoustically isolated quiet booths, and 18 native speakers of English participated in the test. They were presented pairs of synthetic speech with or without the proposed postfilter in random order, and were asked to choose a sample that had better audio quality per pair. Eight sentences randomly chosen from 200 test sentences were read out.

The results of the preference test are shown in Figure 4. The preference score of the DNN-based TTS system with the proposed postfilter was better than that of the system without the filter, and the difference was statistically significant according to t-test ($p<0.01$). Hence, we can conclude that the proposed GAN-based postfilter can improve the quality of synthetic speech in STFT representations.

5. Conclusions

We proposed a learning-based postfilter to reconstruct the high-fidelity spectral texture in STFT spectrograms. To achieve this, we first divide the spectrograms into multiple frequency bands with overlap, reconstruct the individual bands using the GAN-based postfilter trained for each band, and concatenate the bands using the window functions. In the experiments, we applied our postfilter to a DNN-based TTS system. The objective evaluation of the results showed that the proposed postfilter can reproduce fine structures that are close to those of the true data without discontinuity. Moreover, the subjective evaluation showed that our proposed postfilter significantly improves speech quality. For future work, we plan to apply our proposed postfilter to other tasks such as voice conversion and speech enhancement.

Acknowledgements: This work was supported by ACT-I from the Japan Science and Technology Agency (JST), by MEXT/JSPS KAKENHI Grant Numbers 16K16096, JP26730100 and JP26280060, and by The Telecommunications Advancement Foundation Grant.

[1] In synthesizing speech, the phoneme duration is predicted in our acoustic model; therefore, it does not always match the original duration. This means the time duration of the original spectrograms does not exactly match those of the TTS synthesized and GAN postfiltered spectrograms. We manually adjusted them for ease of viewing in the figure.

6. References

[1] T. Toda and K. Tokuda, "A speech parameter generation algorithm considering global variance for HMM-based speech synthesis," *IEICE Trans. Inf. Syst.*, vol. E90-D, no. 5, pp. 816–824, 2007.

[2] H. Silén, E. Helander, J. Nurminen, and M. Gabbouj, "Ways to implement global variance in statistical speech synthesis," in *Proc. Interspeech*, 2012, pp. 1436–1439.

[3] S. Takamichi, T. Toda, G. Neubig, S. Sakti, and S. Nakamura, "A postfilter to modify the modulation spectrum in HMM-based speech synthesis," in *Proc. ICASSP*, 2014, pp. 290–294.

[4] N. Madhu, C. Breithaupt, and R. Martin, "Temporal smoothing of spectral masks in the cepstral domain for speech separation," in *Proc. ICASSP*, 2008, pp. 45–48.

[5] L.-H. Chen, T. Raitio, C. Valentini-Botinhao, J. Yamagishi, and Z.-H. Ling, "DNN-based stochastic postfilter for HMM-based speech synthesis," in *Proc. Interspeech*, 2014, pp. 1954–1958.

[6] K. Hashimoto, K. Oura, Y. Nankaku, and K. Tokuda, "The effect of neural networks in statistical parametric speech synthesis," in *Proc. ICASSP*, 2015, pp. 4455–4459.

[7] T. Kaneko, H. Kameoka, N. Hojo, Y. Ijima, K. Hiramatsu, and K. Kashino, "Generative adversarial network-based postfilter for statistical parametric speech synthesis," in *Proc. ICASSP*, 2017, pp. 4910–4914.

[8] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proc. NIPS*, 2014, pp. 2672–2680.

[9] D. W. Griffin and J. S. Lim, "Signal estimation from modified short-time Fourier transform," *IEEE Trans. Acoust. Speech Sig. Proc.*, vol. 32, no. 2, pp. 236–243, 1984.

[10] S. Takaki, H. Kameoka, and J. Yamagishi, "Direct modeling of frequency spectra and waveform generation based on phase recovery for DNN-based speech synthesis," in *Proc. Interspeech*, 2017.

[11] M. Mirza and S. Osindero, "Conditional generative adversarial nets," in *arXiv preprint arXiv:1411.1784*, 2014.

[12] E. Denton, S. Chintala, A. Szlam, and R. Fergus, "Deep generative image models using a Laplacian pyramid of adversarial networks," in *Proc. NIPS*, 2015, pp. 1486–1494.

[13] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. CVPR*, 2016, pp. 770–778.

[14] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. CVPR*, 2015, pp. 3431–3440.

[15] H. Zen, A. Senior, and M. Schuster, "Statistical parametric speech synthesis using deep neural networks," in *Proc. ICASSP*, 2013, pp. 7962–7966.

[16] D. D. Lee and H. S. Seung, "Algorithms for nonnegative matrix factorization," in *Proc. NIPS*, 2001, pp. 556–562.

[17] P. Smaragdis, B. Raj, and M. Shashanka, "Supervised and semi-supervised separation of sounds from single-channel mixtures," in *Proc. ICA*, 2007, pp. 414–421.

[18] K. Koishida, K. Tokuda, T. Kobayashi, and S. Imai, "CELP coding based on mel-cepstral analysis,," in *Proc. ICASSP*, 1995, pp. 33–36.

[19] S. King and V. Karaiskos, "The blizzard challenge 2011," in *Proc. the Blizzard Challenge workshop 2011*, 2011.

[20] S. Takaki and J. Yamagishi, "A deep auto-encoder based low-dimensional feature extraction from FFT spectral envelopes for statistical parametric speech synthesis," in *Proc. ICASSP*, 2016, pp. 5535–5539.

[21] K. Tokuda, Y. Nankaku, T. Toda, H. Zen, J. Yamagishi, and K. Oura, "Speech synthesis based on hidden Markov models," *Proc. of the IEEE*, vol. 101, pp. 1234–1252, 2013.

[22] V. Nair and G. E. Hinton, "Rectified linear units improve restricted Boltzmann machines," in *Proc. ICML*, 2010, pp. 807–814.

[23] A. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," in *Proc. ICML Workshop*, 2013.

[24] B. Xu, N. Wang, T. Chen, and M. Li, "Empirical evaluation of rectified activations in convolutional network," in *Proc. ICML Workshop*, 2015.

[25] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. ICML*, 2015, pp. 448–456.

[26] D. P. Kingma and M. Welling, "Adam: A method for stochastic optimization," in *Proc. ICLR*, 2015.